A Dictionary Content Management System

Iñaki Alegria, Xabier Arregi, Xabier Artola, Mikel Astiz
Faculty of Computer Science
University of the Basque Country
649 p.k. Donostia
xabier.artola@ehu.es

Leonel Ruiz Miyares
Centro de Lingüística Aplicada
Avda. Raúl Pujols s/n, Parque Zoológico, Vista Alegre
90400, Santiago de Cuba

Abstract
This article presents a new dictionary edition environment, easily adaptable to any data representation. Its main features are user-friendliness, third-party tool integration, configuration flexibility, and Unicode support, among others. Automatically generated entry meta-information is used to provide advanced functionality, such as context-dependent tasks, and any changes are immediately mirrored in a WYSIWYG preview. Its client-server design enables to centrally configure all the system, making it easier to maintain and customize.

1 Introduction

Modern dictionary publishers have to face the need to use advanced computer software. Despite the demanded additional time dedication, it is the only way to maintain complex databases while avoiding usage difficulties. Customized edition environments are necessary but, at the same time, can result very costly.

For that reason, they are mostly developed by big dictionary publishers (McNamara 2003), sometimes from scratch, and their design is kept private. Instead, other approaches have tried to produce non-specific edition and publishing systems, easily adaptable and promisingly independent to the original encoding or storage (Joffe & de Schryver 2004).

With similar goals in mind, we have developed a new flexible content management system, with the aim of building a customizable and scalable open source solution. This demo shows the current state of this dictionary edition environment, which is already being used.

While the main goal was to design a generic sophisticated system, there was the specific need of such an application for the edition of the Cuban *Diccionario Básico Escolar* (DBE) dictionary (Miyares 2003), which we have been working on. This dictionary was originally written and maintained in RTF format but, after a semiautomatic conversion process, it has recently been encoded into XML documents.

Besides, the aim was to integrate multiple dictionaries into a single lexicographic data bank, which involves problematic issues such as redundancy and cross references. In fact, future plans include integrating another dictionary, which would describe the Spanish used in the Caribbean countries, into the database of the DBE, since both dictionaries are closely related in terms of lexicographic content.

2 Features of the CMS

One of the main goals of the edition environment is its ease of use. For that reason, separate design layers have been used to obtain usage transparency and software reusability. Lexicographers working with this application always handle dictionary entries, instead of abstract database values, and should not be aware of technical storage issues. Moreover, WYSIWYG techniques are combined with automatic generation of meta-information, in order to provide accurate context-dependent functionality. To approach this, the following ideas have been taken into account in the design and development of the application.

2.1 Architecture: client-server

Considering the need of a central database, it has been necessary to design a server application that interacts directly with the database, be it relational, XML native, object-oriented or some other. We chose for our system the Berkeley DB XML, which is an open source native XML DB management system. An important fact is that the server must offer a programming interface, in terms of SOAP web services, through which the client application obtains the XML version of entries, indices, etc. The server should also inform of more static aspects of the system as, for example, localization strings.

Besides, the user is allowed to create and edit entries on the client-side, as easy and intuitively as possible. The application offers assistance not only for basic edition operations, but also predefined tasks, queries and integrated consultation resources, such as corpora, third-party dictionaries and spell checkers. On the contrary to the server, the client application always handles XML trees, which will normally be edited and then passed back to the server.

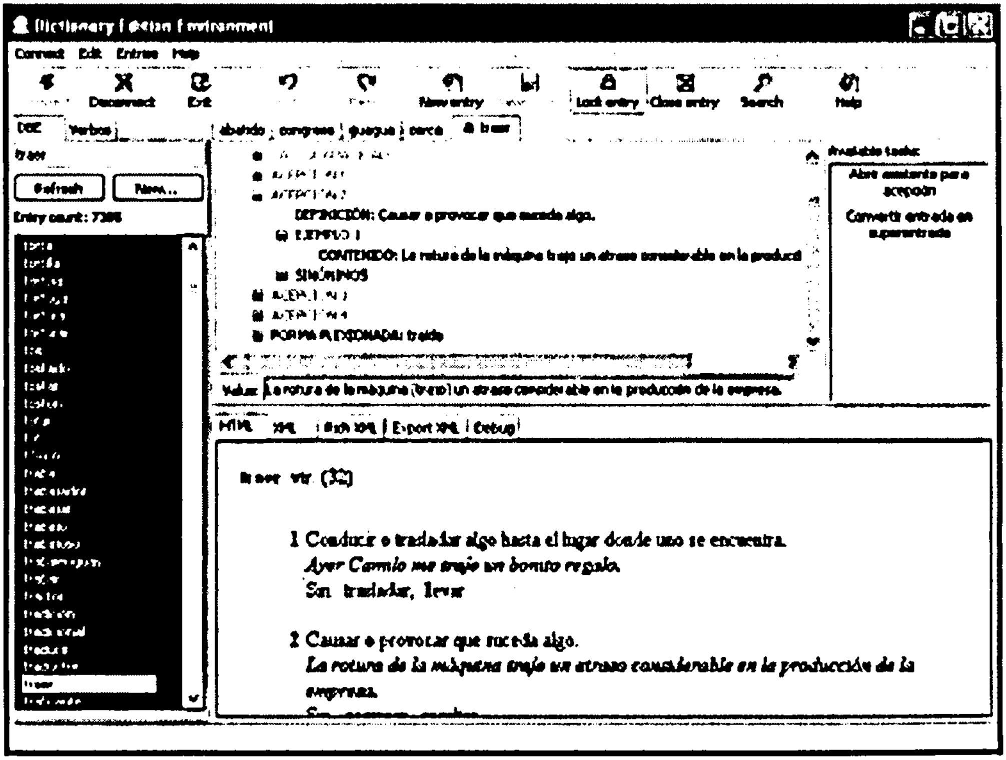
The design of the system enables to easily customize the client by changing some configuration files, which are special purpose XML or XSLT documents stored on the server. This feature makes possible to centrally change the behaviour of the whole system. For instance, the design of dialogs, as well as their sequence order, are completely defined using XSLT scripts, so the client only has to process them without needing any additional embedded information. Other behavioural definitions could likewise be mentioned: search descriptions, graphical entry representation, predefined tasks, templates, hints, localization strings, etc. They are all defined on the server and requested by the client whenever a new session is opened.

2.2 Graphic User Interface (GUI): ease of use

The lexicographers currently using the application, as well as many other traditional publishers, are not used to the complexity these systems entail. For that reason, we consider a requirement for the system to be user-friendly. This edition environment intends to offer a traditional appearance along with advanced functionality, including lock-based concurrency control or user-dependent interfaces.

However, many of the conventional tasks, such as *copy&paste* or formatting operations, are not possible, simply because they do not make sense or because they could be replaced by more sophisticated operations. For example, if the user wants to create a new sense, instead of manually copying and pasting an existing sense, it is more appropriate to use a predefined task that duplicates senses and updates the corresponding data, such as sense numbering.

The graphic interface has been divided into three main parts (see figures 1 and 2): the index, the edition tree and the entry preview.



traer

Entry count: 7386

DEFINICIÓN: Causar o provocar que suceda algo.

EJEMPLO 1

CONTENIDO: La rotura de la máquina trajo un atraso considerable en la producci

SINÓNIMOS

FORMA FLEXIONADA: trajo

Value: La rotura de la máquina trajo un atraso considerable en la producción de la empresa.

traer vtr. (32)

1 Conducir o trasladar algo hasta el lugar donde uno se encuentra.
Ayer Camilo me trajo un bonito regalo.
Sin. trasladar, llevar

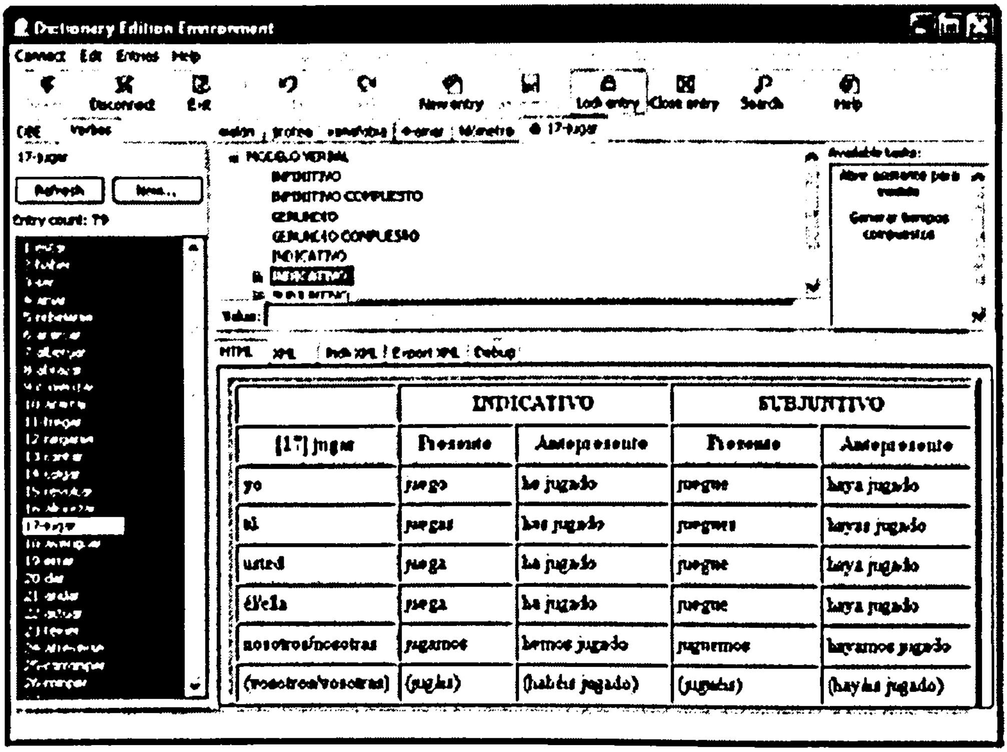
2 Causar o provocar que suceda algo.
La rotura de la máquina trajo un atraso considerable en la producción de la empresa.

Figure 1. GUI of the edition environment: editing an entry.

The index is used to enumerate the different available dictionaries and, of course, their entry list. You can open entries, either new or existing ones, and also perform advanced search queries, each of them conveniently dialog-assisted.

The edition tree is the only component that enables modifying the entry content. Every open entry has an associated tab page and its content is represented, as has been said, in a tree structure, depending on the representation defined by the server. Lexicographers can create, delete or change tree nodes, and they will always have a context-dependent task list. This list is built from the meta-information defined by an intermediate but internal XML tree, which also defines colours, types, permissions, etc.

Any changes made to the entry are immediately shown on screen, following a WYSIWYG strategy, so the lexicographers can always preview the final result. For that reason, at the bottom of the window there is a window that supports HTML, used to properly render the updated entry XML.



	INDICATIVO		SUBJUNTIVO	
[17] jugar	Presente	Antepresente	Presente	Antepresente
yo	juego	he jugado	juegue	haya jugado
tú	juegas	has jugado	juegues	hayas jugado
usted	juega	ha jugado	juegue	haya jugado
él/ella	juega	ha jugado	juegue	haya jugado
nosotros/nosotras	jugamos	hemos jugado	juguemos	hayamos jugado
(vosotros/vosotras)	(jugáis)	(habéis jugado)	(juguéis)	(hayáis jugado)

Figure 2. GUI of the edition environment: editing a verb model.

In addition, every interesting HTML element is hyperlinked and can be used to find the corresponding node in the edition tree. More precisely, when a hyperlinked word is clicked, the associated tree node is found and selected in the edition tree, enabling to immediately perform any changes if wanted. This feature has been proven to be very useful for early users, since the rendering format can be exactly or closely similar to other previous styles. For example, the XSLT stylesheet used in the DBE was easily adapted, just adding some lines, successfully achieving the exact appearance of the electronic version of the dictionary.

2.3 Information flow: flexibility & scalability

Basically, the communication is session-oriented and, depending on the logged user, the server serves the corresponding resources, not only XML documents but also binary files, such as images. The client-server communication is performed using web services or, more precisely, using SOAP services, and can be divided into three main data flows: entry content flow, resource update flow, and control flow.

The first one is responsible for the most dynamic aspect of the dictionary, which of course is the actual content management of the entries. The server gives the last stored XML version of the requested entries and, after the changes made on the client, the updated content is sent back to the server, if necessary. These changes are not accepted unless the entry has been locked previously, so that concurrency problems are avoided in the simplest manner possible.

The resource update flow includes file content transmission and other static aspects. As has been said, behavioural information is described using XML documents, and that involves a considerable amount of data. Therefore, hashing strategies are used to minimize the transferred information, since unchanged files are cached to avoid repeated requests.

The control flow handles control messages, such as session or lock events. It is also used to perform global operations as, for example, search requests or export operations, which affect whole dictionaries instead of being associated to single entries.

3 Conclusion

To sum up, we have developed a flexible client-server application. Lexicographers can easily take advantage of the possibilities it offers and, at the same time, the system administrator can configure most of its aspects by just modifying declarative files. Although it has only been deployed and tested with the DBE, we expect the application will also satisfy the needs of many other potential users.

References

A. Dictionaries

Miyares Bermúdez, E. (dir.) (2003), *Diccionario Básico Escolar.* Centro de Lingüística Aplicada, Santiago de Cuba (Cuba).

B. Other Literature

Joffe, D., de Schryver, G.-M. (2004), 'TshwaneLex: A State-of-the-Art Dictionary Compilation Program'. *Proceedings of EURALEX 2004*, Lorient (France).

McNamara, M. (2003), 'Dictionaries for all: XML to Final Product', *Proceedings of XML Europe 2003 Conference.* London (England).

Alegria, I. et al. (2006), 'Building an Electronic Version of the Cuban Basic School Dictionary'. *Proceedings of EURALEX 2006*, Turin (Italy).